



LL           IIIIII    BBBBBBBB    SSSSSSSS    PPPPPPPP    AAAAAA    WW    WW    NN    NN  
LL           IIIIII    BBBBBBBB    SSSSSSSS    PPPPPPPP    AAAAAA    WW    WW    NN    NN  
LL           IIII    BB    BB    SS    SS    PP    PP    AA    AA    WW    WW    NN    NN  
LL           IIII    BB    BB    SS    SS    PP    PP    AA    AA    AA    WW    WW    NN    NN  
LL           IIII    BB    BB    SS    SS    PP    PP    AA    AA    AA    WW    WW    NNNN    NN  
LL           IIII    BBBBBBBB    SSSSSS    PPPPPPPP    AA    AA    WW    WW    NN    NN    NN  
LL           IIII    BBBBBBBB    SSSSSS    PPPPPPPP    AA    AA    WW    WW    NN    NN    NN  
LL           IIII    BB    BB    SS    PP    AAAAAA    WW    WW    NN    NN    NNNN  
LL           IIII    BB    BB    SS    PP    AAAAAA    WW    WW    NN    NN    NNNN  
LL           IIII    BB    BB    SS    PP    AA    AA    WWWWWW    WWWWWW    NN    NN  
LL           IIII    BB    BB    SS    PP    AA    AA    WWWWWW    WWWWWW    NN    NN  
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS    PP    AA    AA    WW    WW    NN    NN    NN  
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS    PP    AA    AA    WW    WW    NN    NN    NN    NN  
              ....

LL           IIIIII    SSSSSSSS  
LL           IIIIII    SSSSSSSS  
LL           IIII    SS  
LL           IIIIII    SSSSSSSS  
LL           IIIIII    SSSSSSSS

```
1 0001 0 XTITLE 'LIB$SPAWN - Spawn command subprocess'
2 0002 0 MODULE LIB$SPAWN (
3 0003 0 IDENT = '1-004'           ! File: LIB$PAWN.B32 Edit: SR1004
4 0004 0 )
5 0005 1 BEGIN
6 0006 1
7 0007 1 ****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 ****
29 0029 1 .
30 0030 1
31 0031 1 ++
32 0032 1 . FACILITY: General Utility Library
33 0033 1
34 0034 1 . ABSTRACT:
35 0035 1
36 0036 1 . This module contains LIB$SPAWN, a procedure to spawn
37 0037 1 a command subprocess
38 0038 1
39 0039 1 . ENVIRONMENT: User mode - AST reentrant
40 0040 1
41 0041 1 . AUTHOR: Steven B. Lionel, CREATION DATE: 15-Dec-1981
42 0042 1
43 0043 1 . MODIFIED BY:
44 0044 1
45 0045 1 . 1-001 - Original. SBL 15-Dec-1981
46 0046 1 . 1-002 - Remove CLI_NAME parameter. Set EFN to -1 if omitted.
47 0047 1 . Improve comments. SBL 4-Jan-1982
48 0048 1 . 1-003 - Add CLISB_VERSION field to make MMS work with old version.
49 0049 1 . PG 16-Sep-1983
50 0050 1 . 1-004 - Add CLISV_NOKEYPAD, NOTIFY, and NOCONTROL, as well as
51 0051 1 . CLISQ_PROMPT and CLISQ_CLI. SR 13-Dec-1983.
52 0052 1 .--
53 0053 1 .--
```

```
55      0054 1 %SBTTL 'Declarations'  
56  
57      0056 1 : SWITCHES:  
58  
59      0058 1 :  
60      0059 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
61  
62      0060 1 :  
63      0061 1 :  
64      0062 1 : LINKAGES:  
65  
66      0063 1 :  
67      0064 1 :  
68      0065 1 : LINKAGE  
69      C066 1     JSB_ANALYZE_SDESC = JSB (REGISTER=0, REGISTER=1, REGISTER=2) :  
70      0067 1     NOTUSED (3,4,5,6,7,8,9,10,11);  
71  
72      0068 1 :  
73      0069 1 : TABLE OF CONTENTS:  
74  
75      0070 1 :  
76      0071 1 :  
77      0072 1 FORWARD ROUTINE  
78      0073 1     LIB$PAWN;                                ! Spawn command subprocess  
79  
80      0074 1 :  
81  
82      0075 1 :  
83      0076 1 : INCLUDE FILES:  
84  
85      0077 1 :  
86  
87      0078 1 :  
88  
89      0079 1 LIBRARY 'RTLSTARLE';                      ! System symbols, typically from SY$LIBRARY:STARLET.L32  
90  
91      0080 1 :  
92      0081 1 REQUIRE 'RTLIN:RTLPSECT';                  ! Define PSECT declarations macros  
93  
94      0176 1 :  
95      0177 1 REQUIRE 'RTLML:CLIMSG';                  ! CLIS_ messages  
96  
97      0457 1 :  
98      0458 1 :  
99      0459 1 : MACROS:  
100  
101      0460 1 :  
102      0461 1 : MACRO M$VEDESC builds a S-type string descriptor at _TO  
103      0462 1 : which points to the same data area as that pointed to by  
104      0463 1 : the string descriptor at _FROM. NB: since the _TO  
105      0464 1 : descriptor will be used only by SY$CLI which ignores the  
106      0465 1 : DTTYPE and CLASS fields, only the LENGTH and POINTER fields  
107      0466 1 : are built by MOVEDESC.  
108  
109      0467 1 :  
110      M 0468 1 MACRO MOVEDESC (_FROM, _TO) =  
111      M 0469 1 BEGIN  
112      M 0470 1 REGISTER  
113      M 0471 1     RET_STATUS = 0;  
114      M 0472 1     RET_STATUS = LIB$ANALYZE_SDESC R2 ( _FROM;  
115      M 0473 1     BLOCK [-_TO, DSCSW_LENGTH:, .BYTE]  
116      M 0474 1     BLOCK [-_TO, DSCSA_POINTER:, .BYTE] );  
117      M 0475 1     IF NOT .RET_STATUS  
118      M 0476 1     THEN  
119      M 0477 1     RETURN (.RET_STATUS);  
120      M 0478 1     END%;  
121  
122      M 0479 1 :  
123      M 0480 1 :  
124      M 0481 1 : EQUATED SYMBOLS:  
125      M 0482 1 :  
126      M 0483 1 : NONE
```

```
112      0484 1 |  
113      0485 1 | FIELDS:  
114      0486 1 |  
115      0487 1 |     NONE  
116      0488 1 |  
117      0489 1 | PSECTS:  
118      0490 1 |  
119      0491 1 |  
120      0492 1 | DECLARE_PSECTS (LIB);           ! Declare PSECTS for LIB$ facility  
121      0493 1 |  
122      0494 1 |  
123      0495 1 | OWN STORAGE:  
124      0496 1 |  
125      0497 1 |     NONE  
126      0498 1 |  
127      0499 1 | EXTERNAL REFERENCES:  
128      0500 1 |  
129      0501 1 |  
130      0502 1 | EXTERNAL ROUTINE  
131      0503 1 |     LIB$ANALYZE_SDESC_R2: JSB_ANALYZE_SDESC, ! Analyze string descriptor  
132      0504 1 |     SYSSCLI;           ! Request CLI callback  
133      0505 1 |  
134      0506 1 | EXTERNAL LITERAL  
135      0507 1 |     LIB$_INVARG, ! Condition value symbols  
136      0508 1 |     LIB$_NOCLI; ! Invalid argument  
137      0509 1 |           ! No CLI present
```

```
139 0510 1 %SBTTL 'LIB$SPAWN - Spawn command subprocess'
140 0511 1 GLOBAL ROUTINE LIB$SPAWN (
141 0512 1     COMMAND_STRING      : REF BLOCK [, BYTE],
142 0513 1     INPUT_FILE        : REF BLOCK [, BYTE],
143 0514 1     OUTPUT_FILE       : REF BLOCK [, BYTE],
144 0515 1     FLAGS             : REF BLOCK [, BYTE],
145 0516 1     PROCESS_NAME      : REF BLOCK [, BYTE],
146 0517 1     PROCESS_ID        : REF VECTOR [, LONG],
147 0518 1     PROCESS_STATUS    : REF VECTOR [, LONG],
148 0519 1     TERMINATION_EFN   : REF VECTOR [, BYTE],
149 0520 1     TERMINATION_ASTRD, : REF VECTOR [, BYTE],
150 0521 1     TERMINATION_ASPRM, : REF VECTOR [, BYTE],
151 0522 1     PROMPT            : REF BLOCK[, BYTE],
152 0523 1     CLI               : REF BLOCK[, BYTE]
153 0524 1   ) =
154 0525 1
155 0526 1 /**
156 0527 1   FUNCTIONAL DESCRIPTION:
157 0528 1
158 0529 1 LIB$SPAWN requests the calling process's Command Language Interpreter
159 0530 1 to spawn a subprocess for executing CLI commands. It provides
160 0531 1 the same function as the DCL SPAWN command.
161 0532 1
162 0533 1 CALLING SEQUENCE:
163 0534 1
164 0535 1     status.wlc.v = LIB$SPAWN ([command_string.rt.dx]
165 0536 1     [, [input_file.rt.dx] [, [output_file.rt.dx]
166 0537 1     [, [flags.rlu.r] [, [process_name.rt.dx]
167 0538 1     [, [out_pid.wlu.r] [, [process_status.wlc.r]
168 0539 1     [, [termination_efn.rbu.r] [, [termination_astadr.szem.r]
169 0540 1     [, [termination_astprm.rz.v]
170 0541 1     [, [prompt.rt.dx] [, [cli.rt.dx] ]]]]]]]]]])
171 0542 1
172 0543 1 FORMAL PARAMETERS:
173 0544 1
174 0545 1   COMMAND STRING
175 0546 1     A C[] command to be executed by the spawned subprocess,
176 0547 1     passed by descriptor. This is an optional parameter. If omitted,
177 0548 1     commands are taken from the file specified by input-file.
178 0549 1
179 0550 1   INPUT_FILE
180 0551 1     An equivalence name to be associated with the logical name
181 0552 1     SYSS$INPUT in the logical name table for the subprocess, passed
182 0553 1     by descriptor. This is an optional parameter. If omitted,
183 0554 1     the default is the parent terminal.
184 0555 1
185 0556 1   OUTPUT_FILE
186 0557 1     An equivalence name to be associated with the logical name
187 0558 1     SYSS$OUTPUT in the logical name table for the subprocess, passed
188 0559 1     by descriptor. This is an optional parameter. If omitted,
189 0560 1     the default is the parent terminal.
190 0561 1
191 0562 1   FLAGS
192 0563 1     A longword of flag-bits designating optional behavior, passed by
193 0564 1     reference. This is an optional parameter. If omitted, the
194 0565 1     default is that all flags are zero. The flags defined are:
195 0566 1     Bit 0 - NOWAIT
```

196 0567 1 If set, the calling process continues executing in  
197 0568 1 parallel with the subprocess. If clear, the calling  
198 0569 1 process hibernates until the subprocess completes.  
199 0570 1 Bit 1 - NOCLISYM  
200 0571 1 If set, the spawned subprocess does not inherit CLI symbols  
201 0572 1 from its caller. If clear, the subprocess inherits all  
202 0573 1 currently defined CLI symbols.  
203 0574 1 Bit 2 - NOLOGNAM  
204 0575 1 If set, the spawned subprocess does not inherit process  
205 0576 1 logical names from its caller. If clear, the subprocess  
206 0577 1 inherits all currently defined process logical names.  
207 0578 1 Bit 3 - NOKEYPAD  
208 0579 1 If set, keypad symbols and state are passed to subprocess.  
209 0580 1 If not set, the keypad settings are not passed to the subprocess.  
210 0581 1 Bit 4 - NOTIFY  
211 0582 1 If set, causes a message to be broadcast to your terminal  
212 0583 1 when the subprocess completes or aborts. If not set, no  
213 0584 1 message is broadcast. This bit should not be set unless  
214 0585 1 the NOWAIT bit is also set.  
215 0586 1 Bit 5 - NOCONTROL  
216 0587 1 If set, no CR/LF is prepended to any prompt string.  
217 0588 1 If not set, a CR/LF is prepended to any prompt string specified.  
218 0589 1 Bits 6 through 31 are reserved for future expansion and must be zero.  
219 0590 1  
220 0591 1 **PROCESS\_NAME**  
221 0592 1 The name desired for the subprocess, passed by descriptor.  
222 0593 1 This is an optional parameter. If omitted, a unique process  
223 0594 1 name will be generated.  
224 0595 1  
225 0596 1 **OUT\_PID**  
226 0597 1 The longword to receive the process identification of the  
227 0598 1 spawned subprocess, passed by reference. This is an optional  
228 0599 1 parameter. This value is only meaningful if the NOWAIT flags  
229 0600 1 bit is set.  
230 0601 1  
231 0602 1 **PROCESS\_STATUS**  
232 0603 1 The longword to receive the spawned subprocess' final termination  
233 0604 1 status, passed by reference. This is an optional parameter.  
234 0605 1 If the NOWAIT flags bit is set, this value is not stored until the  
235 0606 1 subprocess terminates.  
236 0607 1  
237 0608 1 **TERMINATION\_EFN**  
238 0609 1 The unsigned byte number of a local event flag to be set when the  
239 0610 1 spawned subprocess terminates, passed by reference. This is an  
240 0611 1 optional parameter. If omitted, no event flag is set. Specifying this  
241 0612 1 parameter is only meaningful if the NOWAIT flags bit is set.  
242 0613 1  
243 0614 1 **TERMINATION\_ASTADR**  
244 0615 1 The entry mask of a procedure to be called by means of an AST when  
245 0616 1 the subprocess terminates, passed by reference. This is an optional  
246 0617 1 parameter. Specifying this parameter is only meaningful if the  
247 0618 1 NOWAIT flags bit is set.  
248 0619 1  
249 0620 1 **TERMINATION\_ASTPRM**  
250 0621 1 A value to be passed to the procedure specified by TERMINATION\_ASTADR  
251 0622 1 as an AST routine parameter. Typically, this would be the address of  
252 0623 1 a block of storage to be read or written by the AST procedure. This

253 0624 1 | is an optional parameter. Specifying this parameter is only meaningful  
254 0625 1 | if the NOWAIT flags bit is set and if TERMINATION\_ASTADR has been  
255 0626 1 | specified.  
256 0627 1 |  
257 0628 1 | PROMPT  
258 0629 1 | The prompt string desired for the subprocess, passed by descriptor.  
259 0630 1 | This is an optional parameter. If omitted, the subprocess will use  
260 0631 1 | the same prompt string that the parent uses.  
261 0632 1 |  
262 0633 1 | CLI  
263 0634 1 | The filespecification for the CLI to be run in the subprocess,  
264 0635 1 | passed by descriptor. If you omit the device or directory,  
265 0636 1 | the default is the current device and directory.  
266 0637 1 | This is an optional parameter. If omitted, the subprocess  
267 0638 1 | will use the same CLI as the parent process. If specified,  
268 0639 1 | no context will be copied to the subprocess.  
269 0640 1 |  
270 0641 1 | IMPLICIT INPUTS:  
271 0642 1 |  
272 0643 1 | NONE  
273 0644 1 |  
274 0645 1 | IMPLICIT OUTPUTS:  
275 0646 1 |  
276 0647 1 | NONE  
277 0648 1 |  
278 0649 1 | COMPLETION STATUS:  
279 0650 1 |  
280 0651 1 | SSS\_NORMAL Normal successful completion  
281 0652 1 | SSS\_ACCVIO Access violation  
282 0653 1 | SSS\_DUPLNAME Duplicate process name  
283 0654 1 | LIB\$\_INVARG Invalid argument  
284 0655 1 | LIB\$\_NOCLI No CLI to perform function  
285 0656 1 | fac\$\_xxx Other error trying to create subprocess  
286 0657 1 |  
287 0658 1 | SIDE EFFECTS:  
288 0659 1 |  
289 0660 1 | A command subprocess is spawned. The caller's process hibernates  
290 0661 1 | until the subprocess exits or until an ATTACH back to the calling  
291 0662 1 | process is done, unless NOWAIT is specified.  
292 0663 1 |  
293 0664 1 |--  
294 0665 1 |  
295 0666 2 | BEGIN  
296 0667 2 |  
297 0668 2 | BUILTIN  
298 0669 2 | NULLPARAMETER;  
299 0670 2 |  
300 0671 2 | LOCAL  
301 0672 2 | CLI\_DESC: BLOCK [CLISK\_SRVDESC, BYTE], ! CLI service descriptor  
302 0673 2 | RETURN\_STATUS: BLOCK [4, BYTE];  
303 0674 2 |  
304 0675 2 |  
305 0676 2 | Initialize service request descriptor  
306 0677 2 |  
307 0678 2 |  
308 0679 2 | CHSFILL (0, CLISK\_SRVDESC, CLI\_DESC);  
309 0680 2 | CLI\_DESC [CLISB\_R0TYPE] = CLISR\_CLISERV;

```
310      0681 2   CLI_DESC [CLISW_SERVCOD] = CLISK_SPAWN;  
311      0682 2   CLI_DESC [CLISB_VERSION] = CLISK_SPAWN_VERSION;  
312  
313  
314      0685 2   /*  
315      0686 2   | fill in request descriptor from arguments.  
316      0687 2   |  
317      0688 2   |  
318      0689 2   | CLISO_CMDSTR is descriptor of command string.  
319      0690 2   |  
320      0691 2   | IF NOT NULLPARAMETER (1)  
321      0692 2   | THEN  
322      0693 2   |   MOVEDESC (COMMAND_STRING [0,0,0,0], CLI_DESC [CLISO_CMDSTR]);  
323      0694 2   |  
324      0695 2   |  
325      0696 2   | CLISO_INPUT is descriptor of SYSSINPUT equivalence name.  
326      0697 2   |  
327      0698 2   | IF NOT NULLPARAMETER (2)  
328      0699 2   | THEN  
329      0700 2   |   MOVEDESC (INPUT_FILE [0,0,0,0], CLI_DESC [CLISO_INPUT]);  
330      0701 2   |  
331      0702 2   |  
332      0703 2   | CLISO_OUTPUT is descriptor of SYSSOUTPUT equivalence name.  
333      0704 2   |  
334      0705 2   | IF NOT NULLPARAMETER (3)  
335      0706 2   | THEN  
336      0707 2   |   MOVEDESC (OUTPUT_FILE [0,0,0,0], CLI_DESC [CLISO_OUTPUT]);  
337      0708 2   |  
338      0709 2   |  
339      0710 2   | Currently, only a byte of flags is defined in the SPAWN service  
340      0711 2   | request descriptor. This may be expanded in future versions.  
341      0712 2   |  
342      0713 2   | IF NOT NULLPARAMETER (4)  
343      0714 2   | THEN  
344      0715 3   |   BEGIN  
345      0716 3   |   IF .FLAGS [0,6,26,0] NEQU 0  
346      0717 3   |   THEN  
347      0718 3   |   RETURN LIB$INVARG;  
348      0719 3   |   CLI_DESC [CLISB_FLAGS] = .FLAGS [0,0,8,0];  
349      0720 2   | END;  
350  
351      0722 2   |  
352      0723 2   | CLISO_PRCNAM is descriptor of process name.  
353      0724 2   |  
354      0725 2   | IF NOT NULLPARAMETER (5)  
355      0726 2   | THEN  
356      0727 2   |   MOVEDESC (PROCESS_NAME [0,0,0,0], CLI_DESC [CLISO_PRCNAM]);  
357      0728 2   |  
358      0729 2   |  
359      0730 2   | CLISL_LSTATUS is address of longword where the subprocess' final  
360      0731 2   | termination status is to be stored. If NOWAIT is specified, the  
361      0732 2   | value is not stored until the subprocess actually terminates.  
362      0733 2   |  
363      0734 2   |  
364      0735 2   | IF NOT NULLPARAMETER (7)  
365      0736 2   | THEN  
366      0737 2   |   CLI_DESC [CLISL_LSTATUS] = PROCESS_STATUS [0];
```

```
367 0738 2
368 0739 2
369 0740 2
370 0741 2
371 0742 2
372 0743 2
373 0744 2
374 0745 2
375 0746 2
376 0747 2
377 0748 2
378 0749 2
379 0750 2
380 0751 2
381 0752 2
382 0753 2
383 0754 2
384 0755 2
385 0756 2
386 0757 2
387 0758 2
388 0759 2
389 0760 2
390 0761 2
391 0762 2
392 0763 2
393 0764 2
394 0765 2
395 0766 2
396 0767 2
397 0768 2
398 0769 2
399 0770 2
400 0771 2
401 0772 2
402 0773 2
403 0774 2
404 0775 2
405 0776 2
406 0777 2
407 0778 2
408 0779 2
409 0780 2
410 0781 2
411 0782 2
412 0783 2
413 0784 2
414 0785 2
415 0786 2
416 0787 2
417 0788 2
418 0789 2
419 0790 2
420 0791 2
421 0792 2
422 0793 2
423 0794 2

1+ CLISB_EFN is the number of the local event flag to set upon
2- termination of the subprocess. If -1, no event flag is set.
1- IF NOT NULLPARAMETER (8)
2- THEN CLI_DESC [CLISB_EFN] = .TERMINATION_EFN [0]
1- ELSE CLI_DESC [CLISB_EFN] = -1;

1+ CLISL_ASTADR is the address of the AST routine's entry mask.
2- IF NOT NULLPARAMETER (9)
2- THEN CLI_DESC [CLISL_ASTADR] = .TERMINATION_ASTADR;

1+ CLISL_ASTPRM is the AST routine parameter.
2- IF NOT NULLPARAMETER (10)
2- THEN CLI_DESC [CLISL_ASTPRM] = .TERMINATION_ASTPRM;

1+ CLISQ_PROMPT is descriptor of prompt string.
2- IF NOT NULLPARAMETER (11)
2- THEN MOVEDESC (PROMPT [0,0,0,0], CLI_DESC [CLISQ_PROMPT]);

1+ CLISQ_CLI is descriptor of cli name.
2- IF NOT NULLPARAMETER (12)
2- THEN MOVEDESC (CLI [0,0,0,0], CLI_DESC [CLISQ_CLI]);

1+ Call CLI with request.
2- RETURN_STATUS = SYSSCLI (CLI_DESC);

1+ Adjust error return status, if any.
2- IF NOT .RETURN_STATUS
2- THEN
2- BEGIN
2- IF .RETURN_STATUS EQLU CLIS_INVREQTYP
2- THEN
2- RETURN_STATUS = LIB$_NOCLI;
2- END
2- ELSE RETURN_STATUS = SSS_NORMAL;
```

```

424 0795 2
425 0796 2
426 0797 2      !+ Store output arguments.
427 0798 2
428 0799 2
429 0800 2      IF NOT NULLPARAMETER (6)
430 0801 2      THEN
431 0802 2          PROCESS_ID [0] = .CLI_DESC [CLI$L_OUTPID];
432 0803 2
433 0804 2
434 0805 2      RETURN .RETURN_STATUS;
435 0806 2
436 0807 1      END:           ! End of routine LIB$SPAWN

```

```

.TITLE LIB$SPAWN LIB$SPAWN - Spawn command subprocess
.IDENT \1-004\

```

```

.EXTRN LIB$ANALYZE_SDESC_R2
.EXTRN SYS$CLI, LIB$_INVARG
.EXTRN LIB$_NOCLI

```

```

.PSECT _LIB$CODE,NOWRT, SHR, PIC,2

```

0054	8F	00	56 0000000G	00 007C 000000	.ENTRY LIB\$SPAWN, Save R2,R3,R4,R5,R6 MOVAB LIB\$ANALYZE_SDESC_R2, R6 MOVAB -84(SP), SP MOVCS #0, (SP), #0, #84, CLI_DESC	0511
			5E AC	00 9E 00002 AE 9E 00009 00 2C 0000D 6E 00014		
39	01	6E	05 90 00015	MOVB #5, CLI_DESC MOVW #15, CLI_DESC+1 MOVW #1, CLI_DESC+57	0680	
		AE	0F B0 00018 01 90 0001C 6C 95 00020 16 13 00022			
14	10	50	04 AC 00029 66 16 0002D 51 B0 0002F 52 D0 00033 50 E9 00037 6C 91 0003A	MOVW R1, CLI_DESC+16 MOVL R2, CLI_DESC+20 BLBC RET STATUS, 6\$ CMPB (APT), #2	0681	
				AE		11 13 00027 D5 00024 66 16 0002D 51 B0 0002F 52 D0 00033 50 E9 00037 6C 91 0003A
18	1C	50	08 AC 00044 66 16 00048 51 B0 0004A 52 D0 0004E 50 E9 00052 6C 91 00055	MOVW R1, CLI_DESC+24 MOVL R2, CLI_DESC+28 BLBC RET STATUS, 6\$ CMPB (APT), #3	0682	
				AE		11 13 00042 D5 0003F 66 16 00048 51 B0 0004A 52 D0 0004E 50 E9 00052 6C 91 00055
20	20	50	0C AC 0005F 66 16 00063 51 B0 00065	BLSSU 3\$ TSTL 12(AP) BEQL 3\$ MOVL OUTPUT FILE, R0 JSB LIB\$ANALYZE_SDESC_R2 MOVW R1, CLI_DESC+32	0691	
				AE		11 13 0005D D5 0005A 66 16 00063 51 B0 00065

00	10	BC	24 AE 37 04	52 00 00069 50 E9 00060 6C 91 00070 38: 1A 1F 00073 10 AC 05 00075 15 13 00078 06 ED 0007A 08 13 00080 50 00000000G 8F 00 00082 04 00089	MOVL R2 CLI_DESC+36 BLBC REF STATUS, 6\$ (AP), #4 CMPB 5\$ BLSSU 16(AP) TSTL 5\$ BEQL 4\$ CMPZV #6, #26, @FLAGS, #0 BEQL 4\$ MOVL #LIB\$_INVARG, R0 RET	0713
04	AE	10	BC 90 0008A 48: 05 6C 91 0008F 58: 14 AC 05 00094 11 13 00097 50 14 AC 00 00099 66 16 0009D 51 B0 0009F 52 00 000A3 75 50 E9 000A7 68: 07 6C 91 000AA 78: 0A 1F 000AD 1C AC 05 000AF 05 13 000B2 0C 00 000B4 6C 91 000B9 88: 20 AC 05 000BE 07 13 000C1 38 AE 20 BC 90 000C3 04 11 000C8 38 AE 09 01 8E 000CA 98: 6C 91 000CE 108: 0A 1F 000D1 24 AC 05 000D3 05 13 000D6 30 AE 0A 24 AC 00 000D8 6C 91 000DD 118: 0A 1F 000E0 28 AC 05 000E2 05 13 000E5 34 AE 0B 28 AC 00 000E7 6C 91 000EC 128: 16 1F 000EF 20 AC 05 000F1 11 13 000F4 50 2C AC 00 000F6 66 16 000FA 30 AE 4B 0C 51 B0 000FC 52 00 00100 50 E9 00104 6C 91 00107 138: 16 1F 0010A 30 AC 05 0010C 11 13 0010F 50 30 AC 00 00111 66 1C 00115	MOVW @FLAGS, CLI_DESC+4 (AP), #5 BLSSU 7\$ TSTL 20(AP) BEQL 7\$ MOVL PROCESS_NAME, R0 LIB\$ANALYZE_SDESC_R2 MOVW R1, CLI_DESC+40 MOVL R2, CLI_DESC+44 BLBC RET STATUS, 14\$ (AP), #7 BLSSU 8\$ TSTL 28(AP) BEQL 8\$ MOVL PROCESS_STATUS, CLI_DESC+12 (AP), #8 BLSSU 9\$ TSTL 32(AP) BEQL 9\$ MOVW @TERMINATION_EFN, CLI_DESC+56 BRB 10\$ MNEG B #1, CLI_DESC+56 (AP), #9 BLSSU 11\$ TSTL 36(AP) BEQL 11\$ MOVL TERMINATION_ASTADR, CLI_DESC+48 (AP), #10 BLSSU 12\$ TSTL 40(AP) BEQL 12\$ MOVL TERMINATION_ASTPRM, CLI_DESC+52 (AP), #11 BLSSU 13\$ TSTL 44(AP) BEQL 13\$ MOVL PROMPT, R0 LIB\$ANALYZE_SDESC_R2 MOVW R1, CLI_DESC+60 MOVL R2, CLI_DESC+64 BLBC RET STATUS, 18\$ (AP), #12 BLSSU 15\$ TSTL 48(AP) BEQL 15\$ MOVL CLI, R0 JSB LIB\$ANALYZE_SDESC_R2	0716 0718 0719 0725 0727 0734 0736 0742 0744 0746 0751 0753 0758 0760 0765 0767 0772 0774	

44 AE	51 B0 00117	MOVW	R1, CLI_DESC+68	
48 AE	52 D0 0011B	MOVL	R2, CLI_DESC+72	
30	50 E9 0011F 14\$:	BLBC	REF_STATUS, 18\$	
00000000G 00	5E DD 00122 15\$:	PUSHL	SP	0780
00038822 12	01 FB 00124	CALLS	#1, SYS\$CLI	
00038822 8F	50 E8 00128	BLBS	RETURN_STATUS, 16\$	0786
	50 D1 0012E	CMPL	RETURN_STATUS, #231458	0789
	0C 12 00135	BNEQ	17\$	
50 00000000G	8F D0 00137	MOVL	#LIB\$_NOCLI, RETURN_STATUS	0791
	03 11 0013E	BRB	17\$	0786
50 06	01 D0 00140 16\$:	MOVL	#1, RETURN_STATUS	0794
	6C 91 00143 17\$:	CMPB	(AP), #6	0800
	0A 1F 00146	BLSSU	18\$	
18	AC D5 00148	TSTL	24(AP)	
	05 13 0014B	BEQL	18\$	
18 BC	08 AE D0 0014D	MOVL	CLI_DESC+8, @PROCESS_ID	0802
	04 00152 18\$:	RET		0807

; Routine Size: 339 bytes, Routine Base: \_LIB\$CODE + 0000

; 437 0808 1 !<BLF/PAGE>

LIB\$PAWN  
1-004

LIB\$PAWN - Spawn command subprocess  
LIB\$PAWN - Spawn command subprocess

N 6  
16-Sep-1984 01:15:55 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:39:25 [LIBRTL.SRC]LIB\$PAWN.B32;1

Page 12  
(4)

: 439 0809 1 END  
: 440 0810 1  
: 441 0811 0 ELUDOM

: ! End of module LIB\$PAWN

#### PSECT SUMMARY

Name	Bytes	Attributes
_LIB\$CODE	339	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

#### Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_255\$DUA28:[SYSLIB]STARLET.L32;1	9776	26	0	581	00:00.8

#### COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:LIB\$PAWN/OBJ=OBJ\$:LIB\$PAWN MSRC\$:LIB\$PAWN/UPDATE=(ENH\$:LIB\$PAWN)

: Size: 339 code + 0 data bytes  
: Run Time: 00:13.4  
: Elapsed Time: 00:52.3  
: Lines/CPU Min: 3623  
: Lexemes/CPU-Min: 79045  
: Memory Used: 189 pages  
: Compilation Complete

0210 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

